This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCSVT.2017.2669095, IEEE Transactions on Circuits and Systems for Video Technology

Nonlinear Structural Hashing for Scalable Video Search

Zhixiang Chen, Jiwen Lu, *Senior Member, IEEE,* Jianjiang Feng, *Member, IEEE,* and Jie Zhou, *Senior Member, IEEE*

***Abstract*—In this paper, we propose a nonlinear structural hashing (NSH) approach to learn compact binary codes for scalable video search. Unlike most existing video hashing methods which consider image frames within a video separately for binary code learning, we develop a multi-layer neural network to learn compact and discriminative binary codes by exploiting both the structural information between different frames within a video and the nonlinear relationship between video samples. To be specific, we learn these binary codes under two different constraints at the output of our network: 1) the distance between the learned binary codes for frames within the same scene is minimized, and 2) the distance between the learned binary matrices for a video pair with the same label is less than a threshold and that for a video pair with different labels is larger than a threshold. To better measure the structural information of the scenes from videos, we employ a subspace clustering method to cluster frames into different scenes. Moreover, we design multiple hierarchical nonlinear transformations to preserve the nonlinear relationship between videos. Experimental results on three video datasets show that our method outperforms state-of-the-art hashing approaches on the scalable video search task.**

***Index Terms*—Hashing, scalable video search, neural network, structural information.**

I. Introduction

OVER the past decade, we have witnessed the exponential growth of the video collections on the Internet. In contrast to the rapid growth of video contents, most existing video search engines still rely on textual keyword based indexing, which cannot present all pieces of information in a video and often misses videos due to the absence of text metadata. To leverage the rich visual content of a video, content-based video search uses a set of example videos as queries to retrieve related videos, in which videos are represented by high-dimensional feature representations. While many efforts have been made to improve the accuracy of content based video search [1]–[7], the development of efficient video retrieval technique is still under-explored. Furthermore, video retrieval cannot directly use the text retrieval technique because the extracted video representations are not text-based [8]. Hence, both the tremendous video corpus and high-dimensional video features pose a really important and challenging topic for researchers to develop new search techniques.

As one of the most efficient retrieval methods, the emerging hashing based approximate nearest neighbor search approach has become a popular tool for tackling a variety of large-scale visual analysis problems and has been extensively studied to encode documents or images by a set of short binary codes. Most existing video hashing methods [9]–[12] directly adopt the existing image hashing algorithm to learn a single linear projection matrix to generate binary codes with the goal of preserving similarity between frames. However, different from the imagery data, video clips not only contain many imagery frames but also carry specific structure information, which was ignored in most existing learning-based hashing methods. Therefore, the consistency between frames within the same scene is not encoded in the learned binary representations. Furthermore, such methods can not explicitly encode the nonlinear relationship between videos in the learned binary representations.

To address the abovementioned issues, we propose in this paper a nonlinear structural hashing (NSH) method to learn an efficient neural network for scalable video search. Fig. 1 illustrates the basic idea of our NSH. Specifically, we formulate video hashing as a structure-regularized loss minimization problem to achieve both the video level and the frame level similarity preservations. We design a neural network of multiple hierarchical nonlinear transformations to learn video representations so that inter-class variations of video representations are maximized and intra-class variations of video representations are minimized. Furthermore, we p-reserve the similarity for binary codes of frames within the same scene, which is constructed by employing the subspace clustering method. Specifically, the hashing model is learned by enforcing two constraints on the neural network: 1) the distance of each positive video pair is less than a threshold and that of each negative pair is higher than the threshold, and

This work was supported in part by the National Key Research and Development Program of China under Grant 2016YFB1001001, in part by the National Natural Science Foundation of China under Grant 61672306, Grant 61225008, Grant 61572271, Grant 61527808, Grant 61373074, and Grant 61373090, in part by the National 1000 Young Talents Plan Program, the National Basic Research Program of China under Grant 2014CB349304, in part by the Ministry of Education of China under Grant 20120002110033, and in part by the Tsinghua University Initiative Scientific Research Program. (*Corresponding author: Jiwen Lu.*)

Zhixiang Chen is with Department of Automation, Tsinghua University, Tsinghua National Laboratory for Information Science and Technology (TNList), Beijing, 100084, P. R. China (e-mail: chen-zx10@mails.tsinghua.edu.cn).

Jiwen Lu and Jianjiang Feng are with the Department of Automation, Tsinghua University, State Key Lab of Intelligent Technologies and Systems, Tsinghua National Laboratory for Information Science and Technology (TNList), Beijing, 100084, P. R. China (e-mail: lujiwen@tsinghua.edu.cn; jfeng@tsinghua.edu.cn).

Jie Zhou is with the Department of Automation, Tsinghua University, State Key Lab of Intelligent Technologies and Systems, Tsinghua National Laboratory for Information Science and Technology (TNList), Beijing, 100084, P. R. China and School of Management, University of Shanghai for Science and Technology, Shanghai, 200093, P. R. China (e-mail: jzhou@tsinghua.edu.cn).

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCSVT.2017.2669095, IEEE Transactions on Circuits and Systems for Video Technology

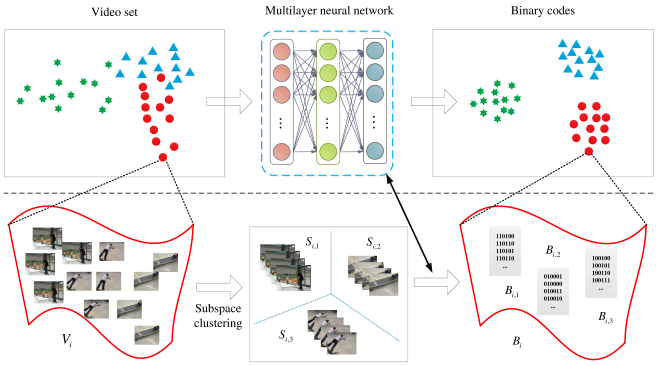


Fig. 1: Illustration of our proposed nonlinear structural hashing method. Firstly, we develop a multi-layer neural network to preserve nonlinear relationship between videos. Secondly, we construct scenes $S_{i,1}, S_{i,2}$ and $S_{i,3}$ from the i-th video V_i through subspace clustering to leverage the statistical knowledge and structural information between frames. Furthermore, similarity constraint is enforced on the binary codes of frames within the same scene to exploit the scene consistency.

2) the variations of the learned binary codes for consecutive frames is minimized. In order to evaluate the performance of the proposed method, we conduct extensive experiments on three large video collections. These datasets contain human activities in unconstrained real-world environments and are challenging for content-based video retrieval. We show the effectiveness of the proposed method and demonstrate that our method achieves significant performance gains compared with previous representative hashing methods for scalable video search.

We summarize the main contributions of this paper as follows:

- We present a hashing learning framework to exploit both the nonlinear relationship between video samples and the structural information between different frames.
- We measure the similarity between video samples based on a set-to-set calculation with frames clustered into different scenes.

The rest of this paper is organized as follows. Section II presents the related work on hashing learning and video retrieval. Section III details our proposed nonlinear structural hashing approach. Section IV presents the experimental results. Finally, Section V concludes this work.

II. Related Work

In this section, we first review several representative learning-based hashing methods and then show some recent progress on video search.

A. Learning-based Hashing

To achieve efficient approximate nearest neighbor search, hashing methods encode high dimensional data as compact binary codes while preserving the similarity of the original data. The hashing methods can be categorized into two classes: data-independent and data-dependent. Data-independent hashing methods adopt random projections to map samples into binary codes. To avoid the recall decrease of long codes for good precision in randomized hash functions [13], data-dependent hashing methods generate compact binary codes by leveraging training sample properties or data label supervision. Existing data-dependent hashing methods can be mainly categorized into two classes: unsupervised and supervised. For the first category, the hashing learning procedure is accomplished without the label information. The binary codes of data samples are learned by employing discrete optimization techniques to preserve the similarity relationship between original high-dimensional features with the goal to be informative. The recent representative algorithms in this category include spectral hashing (SH) [14], iterative quantization (ITQ) [15], Restricted Boltzmann Machines (RBMs) (or semantic hashing) [16], Anchor Graph Hashing (AGH) [17], K-Means Hashing (KMH) [18], Bilinear Projection-based Binary Codes (BPBC) [19], Binary Autoencoder (BA) [20], Sparse Binary Embedding (SBE) [21] and Deep Hashing (DeepH) [22]. While unsupervised hashing is promising to retrieve metric distance neighbors, e.g., ℓ_2 neighbors, the label information is helpful to improve accuracy for searching semantically similar neighbors [23]. In supervised hashing methods, both label information and data properties are utilized to learn hash functions. The hashing model is learned with the goal to minimize the differences between the Hamming affinity over the binary codes and the similarity over the data items, which is determined by the real value features and the data labels. Recent progress in this category has been made in Sequential Projection Learning for Hashing (SPLH) [24], Discriminative Binary Codes (DBC) [25], Minimal Loss Hashing (MLH) [26], Hamming Distance Metric Learning (HDML) [27], LDA Hashing [28], Ranking-Based Supervised Hashing [29], Graph Cuts Coding [30], Kernel-Based Supervised Hashing (KSH) [31], FastHash [32], Supervised Discrete Hashing (SDH) [33], Semisupervised kernel hyperplane learning (SKHL) [34], Semantics-Preserving Hashing (SePH) [35] and Deep Semantic Ranking based Hashing (DSRH) [36].

B. Video Search

Recently, the visual search for video contents has attracted much attention of researchers. The search task can be split into two main phases: feature extraction and search. Feature extraction aims to generate effective, efficient and discriminative representations of videos, and search is to find relevant videos by utilizing the previously extracted features for a given query video. While much research have been devoted into the improvement of search accuracy [1]–[7], [37]–[41], the efficient search algorithm for video search is less studied in the literature. To find relevant videos efficiently, approximate nearest neighbor search is proposed to reduce the complexity of conventional linear scan. Representative efficient approximate nearest neighbor search algorithms include tree-based methods [42]–[45] and quantization methods [8], [46], [47]. However, these methods suffer from the high dimensionality curse and are not suitable for large scale video search. To address this, several learning-based hashing methods have

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCSVT.2017.2669095, IEEE Transactions on Circuits and Systems for Video Technology

been proposed for scalable video search. For example, Cao *et al.* [10] propose to combine feature pooling with hashing for efficient large scale video retrieval. Li *et al.* [9] take advantage of Riemannian Manifold for face video retrieval. Wang *et al.* [12] take account of visual saliency for hash generation. Ye *et al.* [11] exploit the discriminative local visual commonality and temporal consistency to design hash functions. Note that the structure in [11] stands for the spatial structure information within a video frame, while we emphasize the relationship between frames of a video by mentioning structure. While reasonably good performances are demonstrated by these hashing methods, most of them usually learn a single linear transform and ignore the video structural information, which cannot well capture the nonlinear structure of video clips to produce more effective compact hash codes [33]. To this end, we propose a nonlinear structural hashing method by utilizing both the nonlinear relationship between videos and the scene structure, where multi-layer neural network is learned for scalable video search.

III. Structural Video Hashing

In this section, we present a structural hashing model for scalable video search. Assume there are N training videos $\{\boldsymbol{X}_i\}_{i=1}^N$ with category labels $\{y_i\}_{i=1}^N$ where y_i is the label information for video $\boldsymbol{X}_i$. Each video $\boldsymbol{X}_i$ is represented by a collection of n_i sequential frames $\{\boldsymbol{x}_{i,1}, \boldsymbol{x}_{i,2}, \ldots, \boldsymbol{x}_{i,n_i}\} \in \mathbb{R}^{d\times n_i}$. The a-th column of $\boldsymbol{X}_i$, $\boldsymbol{x}_{i,a} \in \mathbb{R}^d$, is the a-th frame of video $\boldsymbol{X}_i$ with a feature length of d. The videos are sampled as representative frames to make the video hashing approach applicable to long videos. The frame sampling method is related to the video summarization [48]–[52]. In this work, we select frames based on uniform sampling. Our goal is to learn an L-bit binary code for each frame. By denoting the binary code of the frame $\boldsymbol{x}_{i,a}$ as $\boldsymbol{b}_{i,a} \in \{-1,1\}^L$, a video $\boldsymbol{X}_i$ with n_i frames can be represented as the binary code matrix $\boldsymbol{B}_i \in \{-1,1\}^{L\times n_i}$.

A. Model

To facilitate efficient video retrieval, the learned binary codes are expected to maintain the local structure of the training videos. First, similar videos are expected to have binary codes with small Hamming distance. We call such similarity between videos as the inter-video similarity. Second, rather than considering video frames as images [11] or video-level feature representation [33], [53], the intra-video similarity is also taken into consideration in the proposed method. The intra-video similarity characterizes the similarity between frames within a video. Thus, the learning objective of the proposed method is to preserve both inter- and intra-video similarities, which is formulated as,

$$\arg\min_{\{\boldsymbol{B}_i\}_{i=1}^N} \mathcal{L} = \mathcal{L}_v + \lambda_1 \mathcal{L}_f, \tag{1}$$

where $\mathcal{L}_v$ and $\mathcal{L}_f$ measure the inter- and intra-video similarity losses, respectively. The parameter λ_1 is the parameter to balance the effects of two kinds of similarity losses.

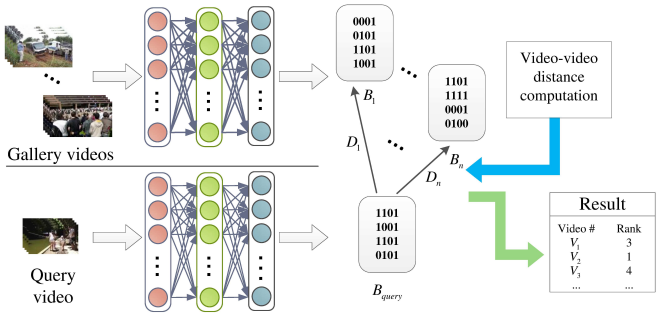


Fig. 2: The workflow of the proposed approach. Firstly, the gallery videos are passed through the learned multilayer neural network to yield the corresponding binary matrices, $\boldsymbol{B}_1, \ldots \boldsymbol{B}_n$, which are stored to constitute the database for the following retrieval. Given a query video, the corresponding binary matrix $\boldsymbol{B}_{query}$ is generated with the learned model. Then, the distances between the binary matrix of query video and those in the database, $D_1, \ldots D_n$, are computed. Finally, a ranking list of database videos is constructed for similarity search.

We define the inter-video similarity loss based on the discriminative distance metric [54] to pursue efficient binary code matrices. The specific form is expressed as,

$$\mathcal{L}_v = \sum_{i=1}^N \sum_{j=1}^N \ell_{i,j}(D(\boldsymbol{B}_i, \boldsymbol{B}_j) - \tau), \tag{2}$$

where $D(\boldsymbol{B}_i, \boldsymbol{B}_j)$ represents the distance between two learned binary code matrices $\boldsymbol{B}_i, \boldsymbol{B}_j$ for videos $\boldsymbol{X}_i, \boldsymbol{X}_j$, which is further defined in Section III-C to leverage the video statistical information. Specifically, (2) aims to seek binary code matrices such that the distance $D(\boldsymbol{B}_i, \boldsymbol{B}_j)$ between $\boldsymbol{X}_i$ and $\boldsymbol{X}_j$ is smaller than a pre-specified threshold τ if $\boldsymbol{X}_i$ and $\boldsymbol{X}_j$ are with the same category ($\ell_{i,j} = 1$), and larger than τ if videos $\boldsymbol{X}_i$ and $\boldsymbol{X}_j$ are with different categories ($\ell_{i,j} = -1$), where the pairwise label $\ell_{i,j}$ denotes the similarity or dissimilarity between a video pair $\boldsymbol{X}_i$ and $\boldsymbol{X}_j$. Note that the value of τ does not influence the minimization of (2) as long as τ is fixed. To make (2) meaningful, τ is assigned a value related to the bit length of the binary code.

The intra-video similarity loss term is to embed the scene consistent constraint between frames of the same scene into the learning objective, which is defined as follows:

$$\begin{aligned}\mathcal{L}_f &= \frac{1}{2}\sum_{i=1}^N \sum_{m=1}^{u_i} \sum_{\boldsymbol{b}_{i,a_1}\in \boldsymbol{S}_{i,m}} \sum_{\boldsymbol{b}_{i,a_2}\in \boldsymbol{S}_{i,m}} \|\boldsymbol{b}_{i,a_1} - \boldsymbol{b}_{i,a_2}\|_2^2 \\ &= \frac{1}{2}\sum_{i=1}^N \left\|\boldsymbol{R}_i \boldsymbol{B}_i^T\right\|_2^2, \end{aligned} \tag{3}$$

where $\boldsymbol{R}_i$ is a constant coefficient matrix related to the scene structure of video $\boldsymbol{X}_i$, $\boldsymbol{S}_{i,m}$ represents one scene of video $\boldsymbol{X}_i$ and u_i is the number of scenes in video $\boldsymbol{X}_i$. Each item $\|\boldsymbol{b}_{i,a_1} - \boldsymbol{b}_{i,a_2}\|_2^2$ is corresponding to a row of $\boldsymbol{R}_i$ with the a_1-th element being 1, the a_2-th element being -1 and the other elements being zeros. Since the scene structure is only related to the content of each video, it is computed before the

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCSVT.2017.2669095, IEEE Transactions on Circuits and Systems for Video Technology

generation of binary codes as will be stated in Section III-C. Here $\|\cdot\|_2$ denotes the matrix L_2-norm. (3) aims to pursue small difference between binary codes for the frames of the same scene in a video, which enforces the smoothly change of binary codes of frames in a scene, and thus explicitly embeds the scene structure into the learned binary codes. This equation introduces the binary code similarity enforcement between frames within the same scene and can be combined with any scene construction method. By combining (2) and (3) together, the frames of the same scene with the same class label are more likely to be mapped to close binary codes.

B. Hashing with Multilayer Neural Network

Given a video frame $\boldsymbol{x}_{i,a}$, a set of hash functions $\boldsymbol{f}_h(\boldsymbol{x}) = [h_1(\boldsymbol{x}), h_2(\boldsymbol{x}), \ldots, h_L(\boldsymbol{x})]$ is learned to produce an L-bit binary code $\boldsymbol{b}_{i,a} \in \{+1, -1\}^L$. A good form of hash functions is critical to obtain the desirable binary codes. In this work, the form of multi-layer neural network is adopted to generate the compact binary codes through multiple nonlinear transformations. Compared to most previous works, which only learn a single linear projection matrix [15], [18], [26], [47], [55], multi-layer neural network can better capture the nonlinear manifold of frames. While the kernel trick can project frames into the kernel space to learn the binary codes, the lack of explicit nonlinear mapping functions usually leads to the scalability problem [31], [56].

As shown in Fig. 2, we develop a network with $K+1$ stacked layers of nonlinear transformations to compute the binary representation $\boldsymbol{b}_{i,a}$ for each frame $\boldsymbol{x}_{i,a}$. Let $p^{(k)}$ be the number of units at the k-th layer, where $1 \leq k \leq K$. The output of the k-th layer is recurrently computed as $\boldsymbol{h}_{i,a}^{(k)} = s(\boldsymbol{W}^{(k)}\boldsymbol{h}_{i,a}^{(k-1)} + \boldsymbol{c}^{(k)}) \in \mathbb{R}^{p^{(k)}}$ with $\boldsymbol{h}_{i,a}^{(1)} = s(\boldsymbol{W}^{(1)}\boldsymbol{x}_{i,a} + \boldsymbol{c}^{(1)}) \in \mathbb{R}^{p^{(1)}}$, where $s(\cdot)$ is a nonlinear activation function, *e.g.*, the $\tanh$ or sigmoid function. $\boldsymbol{W}^{(k)}$ and $\boldsymbol{c}^{(k)}$ are the projection matrix and bias vector to be learned at the k-th layer of the network, respectively. Specifically, the output of the network is calculated as:

$$\boldsymbol{g}_h(\boldsymbol{x}_{i,a}) = \boldsymbol{h}_{i,a}^{(K)} = s(\boldsymbol{W}^{(K)}\boldsymbol{h}_{i,a}^{(K-1)} + \boldsymbol{c}^{(K)}) \in \mathbb{R}^{p^{(K)}}, \quad (4)$$

where the mapping $\boldsymbol{g}_h : \mathbb{R}^d \mapsto \mathbb{R}^{p^{(K)}}$ is a parametric nonlinear function determined by parameters $\{\boldsymbol{W}^{(k)}, \boldsymbol{c}^{(k)}\}_{k=1}^K$. Then, the binary codes are generated by taking the sign of the output of the top layer of the network:

$$\boldsymbol{b}_{i,a} = sgn(\boldsymbol{h}_{i,a}^{(K)}). \quad (5)$$

The binary vectors of all frames in a video can be rewritten as the matrix form $\boldsymbol{B} \in \{-1, +1\}^{L \times n_i}$. On the base of the relationships in (4) and (5), both the parameterized network and a set of binary code matrices of the given training videos can be learned by solving an optimization problem to minimize the predefined loss measurement. As shown in (5), the binary representation of a new video can be obtained by passing each frame through the learned network.

C. Subspace-based Video Distance

The distance between two videos $\boldsymbol{X}_i$ and $\boldsymbol{X}_j$ in the Hamming space is converted to the distance between two sets of binary codes $s_{b_i} := \{\boldsymbol{b}_{i,1}, \boldsymbol{b}_{i,2}, \ldots, \boldsymbol{b}_{i,n_i}\}$ and $s_{b_j} := \{\boldsymbol{b}_{j,1}, \boldsymbol{b}_{j,2}, \ldots, \boldsymbol{b}_{j,n_j}\}$. A naive method is to calculate the difference between the mean values of the two sets s_{b_i} and s_{b_j}. However, it does not take into consideration the set attribute, i.e. the distribution of elements within the set. Considering the scene structure within a video, the frames from a video could be assumed to be distributed on a nonlinear manifold. Hence, a more desirable approach is to measure video distance with the distance between subspaces.

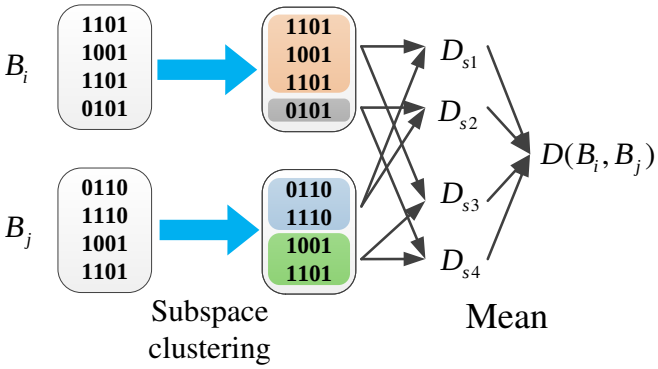


Fig. 3: The computation of distance between binary code matrices of videos. Firstly, the binary matrix of each video $\boldsymbol{B}_i$ is clustered into subspaces to exploit the structural information. Then, the Hamming distances between subspaces of different videos D_{s1}, D_{s2}, D_{s3} and D_{s4} are computed. Finally, the distance between the two binary matrices of videos $D(\boldsymbol{B}_i, \boldsymbol{B}_j)$ is derived on the base of distances between subspaces.

As shown in Fig. 3, to facilitate the computation of the video to video distance, a subspace clustering method is applied for each video, where a nonlinear video representation is decomposed into as a set of linear subspaces. A video $\boldsymbol{X}_i$, which is assumed to come from a low-dimensional manifold, is partitioned into a collection of u_i disjoint subspaces $\{\boldsymbol{S}_{i,1}, \boldsymbol{S}_{i,2}, \ldots, \boldsymbol{S}_{i,u_i}\}$. Each subspace is a linear space spanned by a subset of frames, $\boldsymbol{S}_{i,m} = span(\boldsymbol{X}_{i,m})$, where $\boldsymbol{X}_{i,m} = \boldsymbol{X}_i\boldsymbol{T}_{i,m}$ is a matrix consisting of several columns of $\boldsymbol{X}_i$. That is, $\boldsymbol{X}_i = \bigcup_{m=1}^{k_i} \boldsymbol{X}_{i,m}$ and $\boldsymbol{X}_{i,m} \bigcap \boldsymbol{X}_{i,n} = \varnothing (m \neq n, m, n = 1, 2, \ldots, u_i)$. Each subspace represents a scene in the video, and the procedure of scene clustering is to construct the subspaces for videos. To better construct subspaces for videos we employ the local linear model construction method in [57] to explicitly guarantee the linear property of each subspace. In the one-shot clustering method [57], a seed point is used to generate each new maximal linear patch under the linearity constraint. The maximal linear patch is defined to span a maximal linear subspace. The deviation between the geodesic distances and Euclidean distances in the patch reflects its linear perturbation.

The video to video distance is represented as the integration of distances between pair of subspaces. Typically, a video consists of several clips describing different scenes of an event and similar videos are with a certain overlap in scenes. To classify two videos as the same category, one effective solution is to find the common scenes and measure the similarity of those scenes. Therefore, we define video to video distance

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCSVT.2017.2669095, IEEE Transactions on Circuits and Systems for Video Technology

by the average distance between subspace pair from the two clusters of subspaces as follows:

$$D(\boldsymbol{X}_i, \boldsymbol{X}_j) = \frac{1}{u_i u_j} \sum_{m=1}^{u_i} \sum_{n=1}^{u_j} D_s(\boldsymbol{S}_{i,m}, \boldsymbol{S}_{j,n}), \quad (6)$$

where $D(\boldsymbol{S}_{i,m}, \boldsymbol{S}_{j,n})$ is the distance between two subspaces $\boldsymbol{S}_{i,m}$ and $\boldsymbol{S}_{j,n}$ of videos $\boldsymbol{X}_i$ and $\boldsymbol{X}_j$, respectively. The Hamming distance between the hash codes of frames $\boldsymbol{x}_{i,m,a}$ and $\boldsymbol{x}_{j,n,b}$ of subspaces $\boldsymbol{S}_{i,m}$ and $\boldsymbol{S}_{j,n}$ is adopted to compute the Hamming distance between two subspaces from different videos. In particular, it is computed as the average Hamming distance between frames,

$$D_s(\boldsymbol{S}_{i,m}, \boldsymbol{S}_{j,n}) = \frac{1}{k_{i,m} k_{j,n}} \sum_{a=1}^{k_{i,m}} \sum_{b=1}^{k_{j,n}} d_H(\boldsymbol{x}_{i,m,a}, \boldsymbol{x}_{j,n,b}), \quad (7)$$

where d_H is the Hamming distance of two binary codes and $k_{i,m}$ and $k_{j,n}$ are the numbers of frames in the subspaces $\boldsymbol{S}_{i,m}$ and $\boldsymbol{S}_{j,n}$, respectively. The distance is symmetric and is averaged by the number of frames in subspaces to make the distance less dependent on the size of subspaces.

D. Optimization

Based on the hashing learning function presented in Section III-B and the video distance defined in Section III-C, the objective in (1) can be rewritten as:

$$\begin{aligned} \underset{\{\boldsymbol{W}^{(k)}, \boldsymbol{c}^{(k)}\}_{k=1}^{K}}{\arg\min} \mathcal{L} &= \mathcal{L}_v + \lambda_1 \mathcal{L}_f + \lambda_2 \mathcal{L}_r \\ &= \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} \ell_{i,j} \big(D(\boldsymbol{B}_i, \boldsymbol{B}_j) - \tau \big) \\ &+ \frac{\lambda_1}{2} \sum_{i=1}^{N} \big\| \boldsymbol{R}_i \boldsymbol{B}_i^T \big\|_2^2 \\ &+ \frac{\lambda_2}{2} \sum_{k=1}^{K} \big(\Big\| \boldsymbol{W}^{(k)} \Big\|_2^2 + \Big\| \boldsymbol{c}^{(k)} \Big\|_2^2 \big). \quad (8) \end{aligned}$$

The first two terms $\mathcal{L}_v$ and $\mathcal{L}_f$ enforce inter-/intra-video similarity constraints and are defined in (2) and (3), respectively. These two terms characterize the nonlinear relationship between videos and the scene structure within each video. The last term $\mathcal{L}_r$ is a regularizer to control the scales of the parameters. λ_1 and λ_2 are two parameters to balance the impact of the corresponding terms.

However, the above objective function is intractable due to the discrete constraint introduced by the sgn function to generate binary codes. Following the same signed magnitude relaxation as in [15], [24], we rewrite the objective function as:

$$\begin{aligned} \underset{\{\boldsymbol{W}^{(k)}, \boldsymbol{c}^{(k)}\}_{k=1}^{K}}{\arg\min} \mathcal{L} &= \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} \ell_{i,j} \big(D(\boldsymbol{H}_i^{(K)}, \boldsymbol{H}_j^{(K)}) - \tau \big) \\ &+ \frac{\lambda_1}{2} \sum_{i=1}^{N} \Big\| \boldsymbol{R}_i (\boldsymbol{H}_i^{(K)})^T \Big\|_2^2 \\ &+ \frac{\lambda_2}{2} \sum_{k=1}^{K} \big(\Big\| \boldsymbol{W}^{(k)} \Big\|_2^2 + \Big\| \boldsymbol{c}^{(k)} \Big\|_2^2 \big), \quad (9) \end{aligned}$$

by substituting the binary codes with the output of the deep network $\boldsymbol{H}_i^{(K)}$. Simultaneously, the distance measurement between video subspaces in (7) is redefined as:

$$\begin{aligned} & D_s(\boldsymbol{H}_{i,m}^{(K)}, \boldsymbol{H}_{j,n}^{(K)}) \\ = & \frac{1}{k_{i,m} k_{j,n}} \sum_{a=1}^{k_{i,m}} \sum_{b=1}^{k_{j,n}} (\boldsymbol{h}_{i,m,a}^{(K)} - \boldsymbol{h}_{j,n,b}^{(K)})^T (\boldsymbol{h}_{i,m,a}^{(K)} - \boldsymbol{h}_{j,n,b}^{(K)}) \\ = & \frac{1}{k_{i,m} k_{j,n}} tr\big((\boldsymbol{\Phi}_{i,m}^{(K)} - \boldsymbol{\Psi}_{j,n}^{(K)})^T (\boldsymbol{\Phi}_{i,m}^{(K)} - \boldsymbol{\Psi}_{j,n}^{(K)}) \big), \quad (10) \end{aligned}$$

where $\boldsymbol{\Phi}_{i,m}^{(K)}$ and $\boldsymbol{\Psi}_{j,n}^{(K)}$ are the hidden representations of subspaces $\boldsymbol{H}_{i,m}^{(K)}$ and $\boldsymbol{H}_{j,n}^{(K)}$.

To solve the optimization problem in (9), we use the stochastic sub-gradient descent scheme to obtain the parameters $\{\boldsymbol{W}^{(k)}, \boldsymbol{c}^{(k)}\}_{k=1}^{K}$. The gradients of the objective function $\mathcal{L}$ with respect to the parameters $\boldsymbol{W}^{(k)}$ and $\boldsymbol{c}^{(k)}$ can be computed as follows:

$$\begin{aligned} \frac{\partial \mathcal{L}}{\partial \boldsymbol{W}^{(k)}} &= \sum_{i=1}^{N} \sum_{j=1}^{N} \ell_{i,j} \Bigg(\sum_{m=1}^{u_i} \sum_{n=1}^{u_j} \delta_{i,j}^{m,n} \Big(\Delta_{\boldsymbol{\Phi}}^{(k)} (\boldsymbol{\Phi}_{i,m}^{(k-1)})^T \\ &\qquad - \Delta_{\boldsymbol{\Psi}}^{(k)} (\boldsymbol{\Psi}_{j,n}^{(k-1)})^T \Big) \Bigg) \\ &+ \lambda_1 \sum_{i=1}^{N} \boldsymbol{F}_i^{(k)} \big(\boldsymbol{H}_i^{(k-1)} \big)^T + \lambda_2 \boldsymbol{W}^{(k)}, \quad (11) \end{aligned}$$

$$\begin{aligned} \frac{\partial \mathcal{L}}{\partial \boldsymbol{c}^{(k)}} &= \sum_{i=1}^{N} \sum_{j=1}^{N} \ell_{i,j} \Big(\sum_{m=1}^{u_i} \sum_{n=1}^{u_j} \delta_{i,j}^{m,n} \big(\Delta_{\boldsymbol{\Phi}}^{(k)} - \Delta_{\boldsymbol{\Psi}}^{(k)} \big) \Big) \\ &+ \lambda_1 \sum_{i=1}^{N} \boldsymbol{F}_i^{(k)} + \lambda_2 \boldsymbol{c}^{(k)}, \quad (12) \end{aligned}$$

where the coefficient $\delta_{i,j}^{m,n} = \frac{1}{u_i u_j} \cdot \frac{1}{k_{i,m} k_{j,n}}$ for the subspace pair $\{\boldsymbol{H}_{j,n}^{(K)}, \boldsymbol{H}_{i,m}^{(K)}\}$.

The updating equations are computed as follows:

$$\Delta_{\boldsymbol{\Phi}}^{(K)} = (\boldsymbol{\Phi}_{i,m}^{(K)} - \boldsymbol{\Psi}_{j,n}^{(K)}) \odot s'(\boldsymbol{Z}_{\boldsymbol{\Phi}_{i,m}}^{(K)})\big), \quad (13)$$

$$\Delta_{\boldsymbol{\Psi}}^{(K)} = (\boldsymbol{\Phi}_{i,m}^{(K)} - \boldsymbol{\Psi}_{j,n}^{(K)}) \odot s'(\boldsymbol{Z}_{\boldsymbol{\Psi}_{j,n}}^{(K)})\big), \quad (14)$$

$$\boldsymbol{F}_i^{(K)} = \big(\boldsymbol{R}_i^T \boldsymbol{R}_i (\boldsymbol{H}_i^{(K)})^T \big)^T \odot s'(\boldsymbol{Z}_i^{(K)}), \quad (15)$$

$$\Delta_{\boldsymbol{\Phi}}^{(k)} = \big((\boldsymbol{W}^{(k+1)})^T \Delta_{\boldsymbol{\Phi}}^{(k+1)} \big) \odot s'(\boldsymbol{Z}_{\boldsymbol{\Phi}_{i,m}}^{(k)}), \quad (16)$$

$$\Delta_{\boldsymbol{\Psi}}^{(k)} = \big((\boldsymbol{W}^{(k+1)})^T \Delta_{\boldsymbol{\Psi}}^{(k+1)} \big) \odot s'(\boldsymbol{Z}_{\boldsymbol{\Psi}_{j,n}}^{(k)}), \quad (17)$$

$$\boldsymbol{F}_i^{(k)} = \big((\boldsymbol{W}^{(k+1)})^T \boldsymbol{F}_i^{(k+1)} \big) \odot s'\big(\boldsymbol{Z}_i^{(k)}\big), \quad (18)$$

where $k = 1, 2, \ldots, K-1$. Here the operation $\odot$ denotes the element-wise multiplication, and $\boldsymbol{Z}_{\boldsymbol{\Phi}_{i,m}}^{(k+1)} = \boldsymbol{W}^{(k+1)} \boldsymbol{\Phi}_{i,m}^{(k)} + \boldsymbol{c}^{(k+1)}$, $\boldsymbol{Z}_{\boldsymbol{\Psi}_{j,n}}^{(k+1)} = \boldsymbol{W}^{(k+1)} \boldsymbol{\Psi}_{j,n}^{(k)} + \boldsymbol{c}^{(k+1)}$, $\boldsymbol{Z}_i^{(k+1)} = \boldsymbol{W}^{(k+1)} \boldsymbol{H}_i^{(k)} + \boldsymbol{c}^{(k+1)}$.

The gradient descent algorithm is adopted to update the parameters of the network until convergence, and the specific updating rule is as follows:

$$\boldsymbol{W}^{(k)} = \boldsymbol{W}^{(k)} - \eta \frac{\partial \mathcal{L}}{\partial \boldsymbol{W}^{(k)}}, \quad (19)$$

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCSVT.2017.2669095, IEEE Transactions on Circuits and Systems for Video Technology

Algorithm 1: NSH

Input: Training videos $\boldsymbol{X}$ and label, iterative number R, learning rate η, network layer number K, parameters λ_1, λ_2, and convergence error ε.
Output: Network parameters $\{\boldsymbol{W}^{(k)}, \boldsymbol{c}^{(k)}\}_{k=1}^{K}$.
Initialization: Construct subspaces for each video; initialize $\{\boldsymbol{W}^{(k)}, \boldsymbol{c}^{(k)}\}_{k=1}^{K}$ by getting the top $p^{(1)}$ eigenvectors.
for $r = 1, 2, \ldots, R$ **do**
  for $i = 1, 2, \ldots, N$ **do**
    Initially setting $\boldsymbol{H}^{(0)}(\boldsymbol{X}_i) = \boldsymbol{X}_i$.
  end
  for $k = 1, 2, \ldots, K$ **do**
    for $i = 1, 2, \ldots, N$ **do**
      Compute $\boldsymbol{H}^{(k)}(\boldsymbol{X}_i)$ according to (4).
    end
  end
  for $k = K, K-1, \ldots, 1$ **do**
    Calculate the gradients with (11) and (12).
  end
  for $k = 1, 2, \ldots, K$ **do**
    Update $\boldsymbol{W}^{(k)}$ and $\boldsymbol{c}^{(k)}$ according to (19) and (20).
  end
  Calculate $\mathcal{L}$ using (9).
  If $r > 1$ and $|\mathcal{L}_r - \mathcal{L}_{r-1}| < \varepsilon$, go to **Return**.
end
Return: $\{\boldsymbol{W}^{(k)}, \boldsymbol{c}^{(k)}\}_{k=1}^{K}$.

$$\boldsymbol{c}^{(k)} = \boldsymbol{c}^{(k)} - \eta \frac{\partial \mathcal{L}}{\partial \boldsymbol{c}^{(k)}}, \tag{20}$$

where η is the learning rate.

The step by step description of the proposed NSH approach is provided in **Algorithm 1**.

E. Implementation Details

To apply our approach for video search, similar videos are retrieved by the Hamming distance. Specifically, the binary codes of the database videos are generated along with the scene clustering offline. Given a query video, all video frames are firstly used to construct the frame set. These frames are further processed to generate binary codes by using the learned hashing model and accomplish scene clustering via the subspace construction method. With the binary codes and scene structure of the query video and the database videos, the distance between query video and every database video can be computed according to (6) and (7). Then, a ranking list of database videos can be constructed for similarity search.

F. Discussion

In this subsection, we highlight the difference between our nonlinear scalable video retrieval approach and some recently proposed hashing methods.

1) Nonlinear Scalable Hashing [22], [32]: Some recently proposed hashing approaches harness the nonlinear manifold structures of data samples and have achieved superior performance [22], [32]. Currently, these methods are designed for the retrieval on imagery datasets. Such methods only consider the similarity relationship between data samples, i.e., image-to-image or video-to-video similarity. However, videos contain rich structure information at each data point compared to imagery data. Different from these methods, our video hashing method exploits both the relationship between videos and the structure information in each video to learn binary codes.

2) Video Hashing for Retrieval [11]: Ye *et al.* [11] propose to learn effective hash functions by exploring the structure learning techniques and have demonstrated promising results. Their work exploits the common local visual patterns in video frames with the same class label, together with the temporal consistency over consecutive frames, to learn hash functions. In contrast, our hashing learning approach leverages both the nonlinear relationship between videos and the scene consistency between frames within a video to learn discriminative binary codes. Hence, both the relationship between videos and the scene structure are exploited.

IV. Experiments

To evaluate the effectiveness of our proposed nonlinear structural hashing (NSH) method, we perform extensive experiments on three large video collections, i.e., Columbia Consumer Video (CCV) [58], YLI Multimedia Event Detection (YLI-MED) [59], [60] and ActivityNet [61] datasets, and compare our method with several state-of-the-art methods. These datasets are challenging for retrieval due to the presence of large intra-class variation between videos and the motions in the wild between frames. The following subsections describe the details of the experiments and results.

A. Experimental Settings

We compare our NSH with several representative linear hashing methods. Specifically, we take Iterative Quantization (ITQ) [15] and Canonical Correlation Analysis Iterative Quantization (CCA-ITQ) [15] as baselines. These methods include both unsupervised (ITQ) and supervised (CCA-ITQ) hashing paradigms. Since our NSH method is nonlinear, we also compare it with four state-of-the-art nonlinear hashing methods, Deep Hashing (DeepH) [22], Supervised Deep Hashing (SDeepH) [22], Kernel-Based Supervised Hashing (KSH) [31] and Supervised Discrete Hashing (SDH) [33]. The source codes of these baseline methods were kindly provided by the authors. We use the suggested parameters of these methods from the corresponding authors. While several video hashing methods are recently presented in the literature [10], [11], [62], most of them adopt data-independent techniques, and require specific settings, such as multiple-feature extraction [62] or submodular representation [10]. Therefore, they are not considered as comparable baselines. We followed the widely used evaluation protocols in [11], [24], i.e., Hamming ranking and hash lookup, and adopted the following two evaluation metrics for consistent evaluations across different methods: 1) mean average precision (mAP), which represents the overall performance of different hashing methods by the area under the precision-recall curve; and 2) Hamming look-up result with the hamming radius set as r, which measures the precision

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCSVT.2017.2669095, IEEE Transactions on Circuits and Systems for Video Technology

over all retrieved samples that fall into the buckets within a set hamming radius, i.e., $r = 2$. We followed [31] to set zero precision for conditions failing to find any hash bucket for a query, different from previous work which computed the mean hash lookup precision over all queries by ignoring the failed queries. As stated in [62], mAP has been demonstrated to have especially good discrimination and stability. The nearest video sequences are returned by computing and ranking the Hamming distance between query video and gallery videos according to (6). For the calculation of Hamming look-up precision, we followed the protocol in [11]. The ground-truth relevant instances for a query are defined as those sharing at least one category label with it.

For our NSH method, we trained our deep model with a 3-layer model with the dimensions of each layer being 512, 200, L, to produce L-bit long binary codes. We set the value of τ to be L/nC with nC as the number of category of training samples. We set parameters λ_1 and λ_2 to 0.1 and 0.001, respectively, through 5 fold cross validation with one fifth of the training set as validation set. We optimized our hashing model with 5 iteration updates. In our models, we adopted the hyperbolic tangent function as the nonlinear activation function. In addition, for all the supervised algorithms, we randomly select a small subset of labeled samples to train the learned hash functions on all three datasets. For a fair comparison, we trained NSH and supervised baselines on the same set of training data. During batch training for the image-based hashing methods, we randomly selected frames from different videos to ensure large diversity of the training samples.

B. Experimental Comparison on CCV

The Columbia Consumer Video (CCV) dataset [58] has been widely adopted in several recent studies on video hashing. There are 9,317 videos gathered from YouTube, which covers 20 categories annotated based on Amazon's MTurk platform. On average, the video duration is around 80 seconds. Interested readers are referred to [58] for more details. We followed the experimental settings in [11] for our experiments. Specifically, we randomly selected 20, 20 and 100 videos per category to construct the training, query and gallery sets. We sampled each video every 2 seconds to extract frames and ensure a minimum of 30 frames for each video.

1) Results on the Hand-Crafted Feature: The SIFT descriptors over key points on each frame are extracted with two different key point detectors, i.e., Different of Gaussian [63] and Hessian Affine [64]. The 128-dimensional SIFT features are then fed into the quantization procedure to derive 5,000-dimensional BoW representation [65] for each frame. Fig. 4(a) presents the comparison between NSH and other baseline methods in terms of mAP on the CCV dataset with respect to different number of bits. We see that our NSH method outperforms all other methods in all bit numbers, including the previous state-of-the-art SDeepH and DeepH. We see that the supervised methods generally perform better than unsupervised methods. Moreover NSH shows superior performance compared to SDeepH. This is because both the video label information and the frame level label information are important to the learning of binary codes. Furthermore, the nonlinear hashing methods such as DeepH, SDeepH and NSH show better performance than other linear methods. The reason is that the nonlinear relationship between samples is preserved. However, the KSH method surprisingly delivers low performance in our results. This may be due to that the provided kernel model does not fit the video samples well. We also see that the results of SDH increase with longer bit length, which is consistent with the importance of minimization quantization for long binary codes. Fig. 4(b) presents the comparison on results in terms of the precision of first 500 retrieved samples. We see that our method is superior to the best competitor. The precisions of ITQ and CCA-ITQ increase for longer binary codes, which implies that reducing the quantization error is a reasonable objective. In Fig. 4(c), we show the precision recall curves for different methods when the length of binary codes is 32 bits. We see that our method presents the best performance. This is because that the video structure is exploited in our hashing learning model.

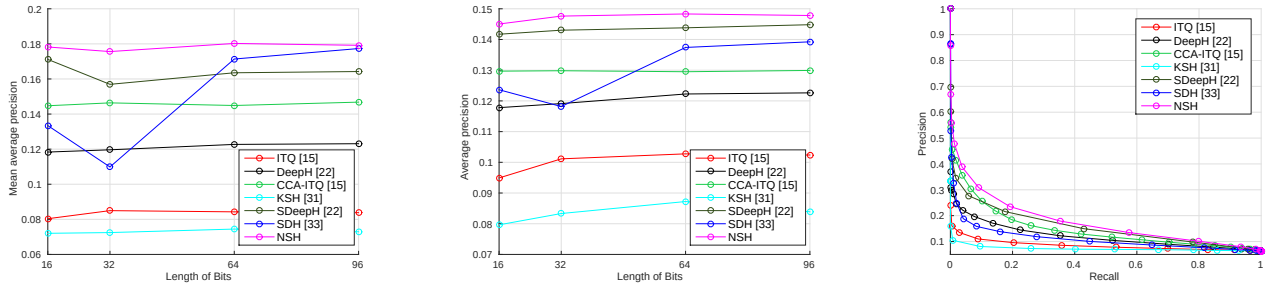
2) Results on the CNN Feature: Besides hand-crafted bag-of-SIFT feature, based on SIFT descriptors, we also evaluated our NSH with the state-of-the-art deep Convolutional Neural Network (CNN) features. To extract the representation for each frame, we adopted the VGG model [66] pretrained on the ImageNet [67], and used the output of the layer 'fc7' as suggested in [68]. Fig. 5 and 6 summarize the performance of different hashing methods. As expected, the overall performances are higher than those on the hand-crafted features for both the baseline methods and our NSH. In Fig. 5(a), we see that NSH outperforms all the competitors by a large margin in terms of the mAP. Note that the substitution of hand-crafted features with CNN features enlarges the gaps between different hashing methods, which indicates a well designed hashing method is important to capture the similarity structure of original features. The results in Fig. 5(b) show the precisions of top 500 retrieved samples for different hashing methods. Fig. 5(c) presents the comparison on precision within Hamming radius 2, where our method is competitive to the remaining hashing methods. However, the precisions of most methods drop with longer binary codes. The reason is that longer codes result in lower probability that samples fall in the same bucket.

Besides the performance, we also compare the testing time of our NSH method with those of other baseline methods. In Table I, we report the testing time of all involved methods for various lengths of binary codes on the CCV dataset. The time is measured in seconds over all testing videos. The computing platform is equipped with 4.0GHz Intel CPU and 32GB RAM. From the table, we can see that the testing time consumed by our NSH method is comparable to those of existing methods.

C. Experimental Comparison on YLI-MED

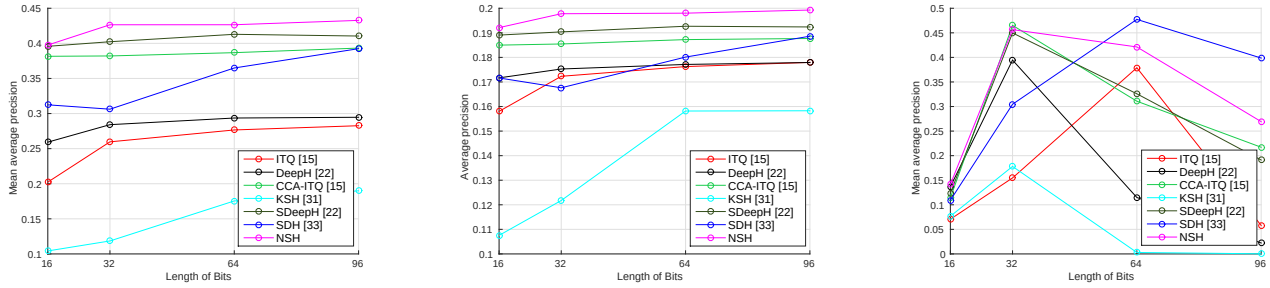
The YLI-MED dataset [59] is the video subset of the Yahoo Flickr Creative Commons 100 Million (YFCC100M) dataset specialized for Multimedia Event Detection research, which is similar to the existing TRECVID MED's HAVIC corpus. In particular, the 10 current available annotated events in YLI-MED, such as Birthday, Wedding and Woodworking, were

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCSVT.2017.2669095, IEEE Transactions on Circuits and Systems for Video Technology



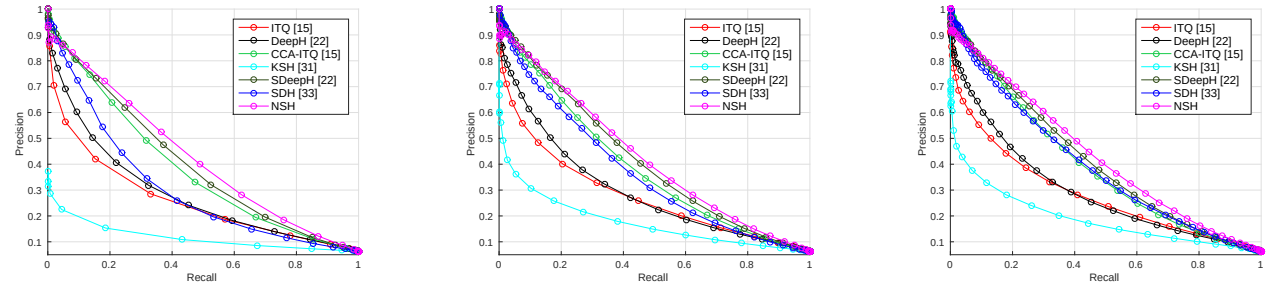
(a) mean Average Precision (b) Precision @500 (c) Precision-recall @32bit

Fig. 4: The results of different hashing methods on the CCV dataset with hand-craft features.



(a) mean Average Precision (b) Precision @500 (c) Precision within Hamming radius 2

Fig. 5: The comparison of different hashing methods on the CCV dataset with CNN features.



(a) 32 bits (b) 64 bits (c) 96 bits

Fig. 6: The comparison of precision-recall curves for different hashing methods on the CCV dataset with binary codes of different lengths.

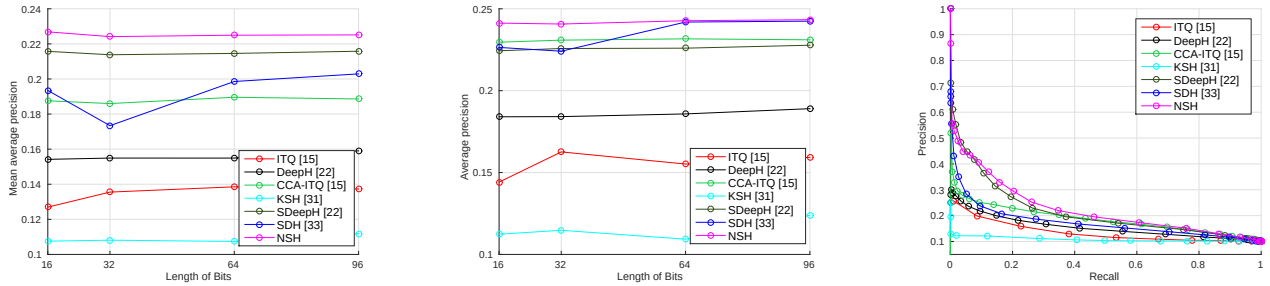
TABLE I: Testing time in seconds of different hashing methods on the CCV dataset.

Methods	length of binary code			
	16	32	64	96
ITQ [15]	4.63	5.12	4.94	5.41
DeepH [22]	4.42	4.95	5.53	5.99
CCA-ITQ [15]	19.95	19.99	19.95	19.79
KSH [31]	16.38	17.15	16.52	16.66
SDeepH [22]	32.43	33.65	32.42	32.77
SDH [33]	92.75	97.13	97.49	95.99
NSH	18.71	19.22	19.03	19.18

included in the TRECVID MED 2011 evaluation run by the National Institute of Standards and Technology (NIST). YLI-MED is one of the largest public available video collections with manual annotation. A subset of the videos in the YFC-C100M is adopted for the MED task in the TRECVID benchmark organized by NIST. There are around 2000 annotated videos in the YLI-MED dataset. The video duration varies from 2 to 200 seconds with average value around 40 seconds. Here, we selected frames every second and ensure at least 30 frames for each video. We randomly selected 20 and 20 videos in each category to construct the training set and query set, respectively. We selected another 100 videos each category as gallery set for retrieval.

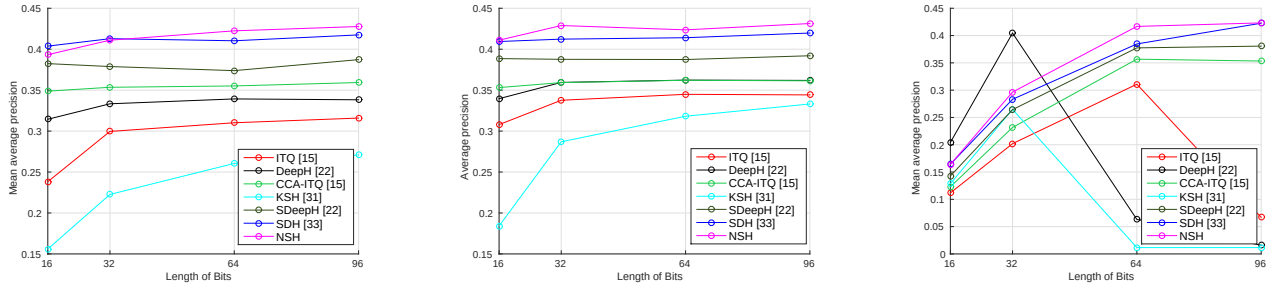
1) Results on the Hand-Crafted Feature: For each frame, we followed the same feature extraction procedure as in the experiments on the CCV dataset. In Fig. 7(a), we show the mAP results of NSH as well as other representative hashing methods. We see that out method outperforms other methods by a large margin. Similar with the results on the CCV dataset, the performance of NSH is better than those of image-based supervised methods and linear projection based methods. This is because that both the video structure and the nonlinear

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCSVT.2017.2669095, IEEE Transactions on Circuits and Systems for Video Technology



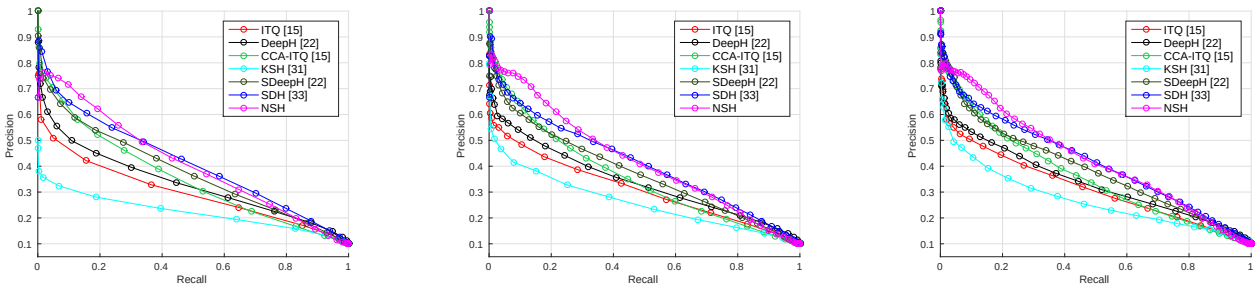
(a) mean Average Precision (b) Precision @100 (c) Precision-recall @32bit

Fig. 7: The results of different hashing methods on the YLI dataset with hand-crafted features.



(a) mean Average Precision (b) Precision @100 (c) Precision within Hamming radius 2

Fig. 8: The comparison of different hashing methods on the YLI dataset with CNN features.



(a) 32 bits (b) 64 bits (c) 96 bits

Fig. 9: The comparison of precision-recall curves for different hashing methods on the YLI dataset with binary codes of different lengths.

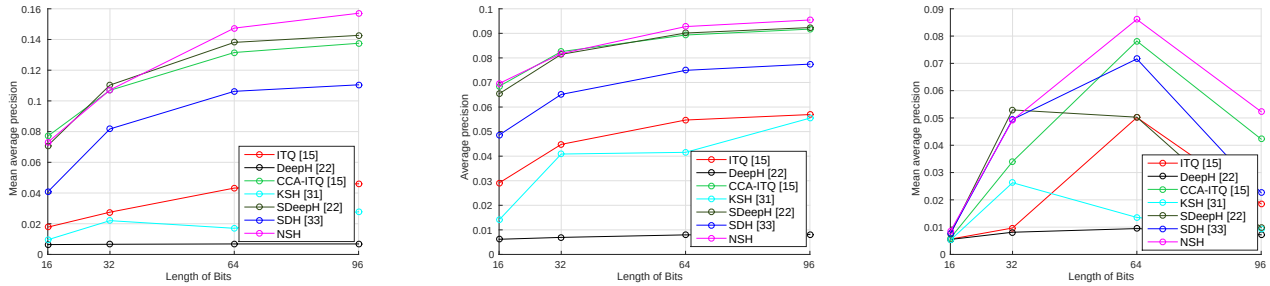
transformation play an important role to map samples into binary codes. In Fig. 7(b), we show that our NSH delivers better performance than the competitors, which is similar to the results on CCV dataset. Fig. 7(c) shows the precision-recall curves with binary codes of 32 bits, which is the detailed presentation of the mAP. The results show that our NSH method delivers the best retrieval performance.

2) Results on the CNN Feature: Similar to the CCV dataset, we extracted the CNN feature for each sampled frame in YLI dataset to test the retrieval performance. Fig. 8(a) and (b) show the mAP and precision of top 100 retrieved samples for each hashing method, respectively. We see that the CNN feature enhances the performance. Nevertheless, our NSH method is still comparable to the best competitor. Similar to the results on the CCV dataset, the performance of SDH is slightly better than our NSH due to the careful optimization of the quantization loss. In Fig. 8(c), we show the results of Hamming lookup precision within radius 2. We see that our method produces comparable precision to other representative hashing methods. In Fig. 9, we show the precision-recall curves of different hashing methods for different lengths of binary codes. These results clearly show the superiority of our NSH method.

D. Experimental Comparison on ActivityNet

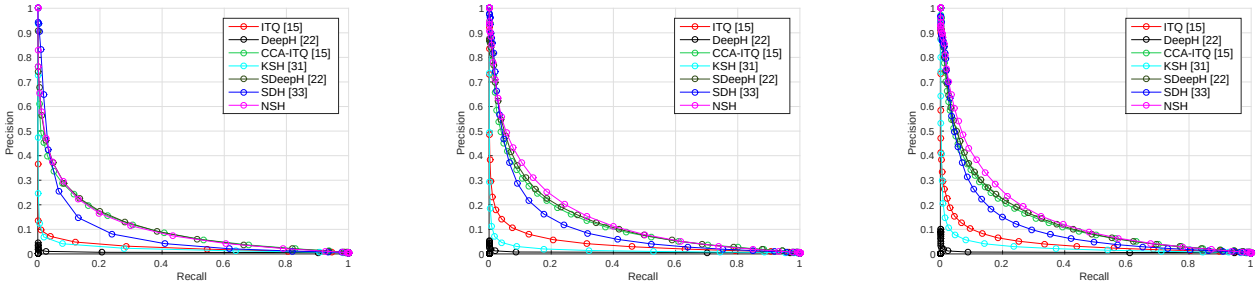
We also evaluate our approach on a large-scale video benchmark dataset, ActivityNet. The ActivityNet dataset [61] is recently released for human activity recognition and understanding. ActivityNet consists of around 20,000 video clips in 200 classes with 10,024 training, 4,926 validation and 5,044 testing videos, totaling 648 hours of video. Compared to the above two datasets, ActivityNet is more challenging, since it contains more videos, more classes of fine-grained actions, has greater intra-class variance and consists of longer, untrimmed

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCSVT.2017.2669095, IEEE Transactions on Circuits and Systems for Video Technology



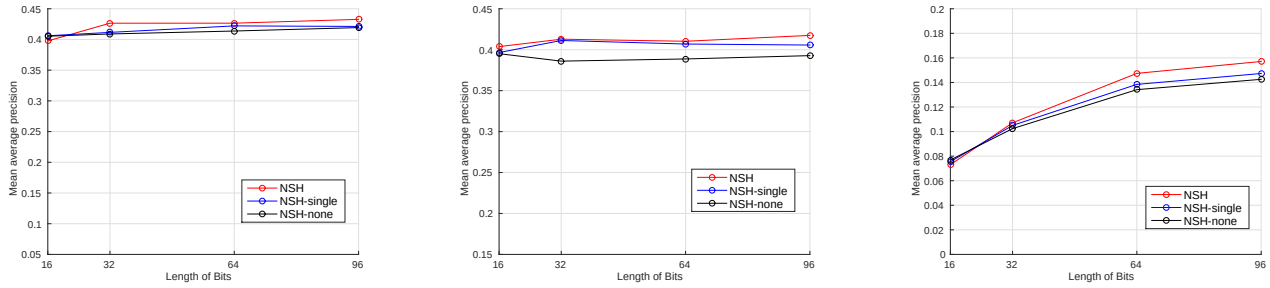
(a) mean Average Precision (b) Precision @100 (c) Precision within Hamming radius 2

Fig. 10: The comparison of different hashing methods on the AcitvityNet dataset with CNN features.



(a) 32 bits (b) 64 bits (c) 96 bits

Fig. 11: The comparison of precision-recall curves for different hashing methods on the ActivityNet dataset with binary codes of different lengths.



(a) CCV (b) YLI (c) AcitivityNet

Fig. 12: Scene structure evaluations on CCV, YLI and ActivityNet datasets.

videos. The experimental setting for this dataset is the same as that on YLI-MED dataset. Since the authors of ActivityNet have not released the test set, we use the validation set as our query set, take 30 videos per categories as the training set and use the remaining videos as the gallery set. For each frame, we followed the same feature extraction procedure as in the experiments on two previous datasets. Because the performance with hand-crafted feature on this dataset is too low, we only show the results with CNN feature.

In Fig. 10(a), we show the mAP values for different hashing methods. We see that the performance is much lower than those on the CCV and YLI datasets due to the greater intra-class variation, more videos, more classes and untrimmed videos. Nevertheless, our method still ranks top compared with the baseline methods. In Fig. 10(b), we show the precisions of top 100 retrieved samples, where our method is competitive to other hashing methods. Fig. 10(c) shows the results of Hamming lookup precision within radius 2. We see that our NSH achieves the best performance. To provide a detailed observation of the retrieval performance, we show the precision-recall curves for 32, 64 and 96 bits binary codes in Fig. 11.

TABLE II: MAP comparison of different video hashing methods on the CCV dataset.

	16	32	48	64
SH [10], [14]	7.7	7.8	7.8	7.8
SVH [11]	11.2	13.0	14.6	15.8
NSH	**12.7**	**13.1**	**15.2**	**16.1**

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCSVT.2017.2669095, IEEE Transactions on Circuits and Systems for Video Technology

E. Comparison with Existing Video Hashing Methods

In order to give a quantitative performance comparison with existing video hashing methods, we followed the experimental settings in [11] to evaluate the performance of our NSH. Here, we compared our NSH with the spectral hashing used in [10], [14] and the structure learning-based video hashing [11]. In Table II, we show the mAP values of different hashing methods for different lengths of binary codes. We see that our NSH achieves competitive results compared to SVH. The reason is that the learned neural network model encodes both the nonlinear relationship between videos and the scene consistency between frames within a video.

F. Effect of Scene Structure

To evaluate the effectiveness of scene structure, we compared the comprehensive mAP metric across different hashing methods to present an overall measurement of retrieval performance. Fig. 12(a), (b) and (c) present the results on CCV, YLI and ActivityNet with CNN features. We compared NSH with NSH-single, which considers the whole video as a single subspace, and NSH-none, which ignores the scene structure by setting λ_1 in (8) as 0. These comparisions show that our NSH consistently achieves better retrieval performance. Note that the scene is related to the video content and is constructed on the base of the original features. Therefore, the scene information of each video is precomputed and stored off-line. Furthermore, the scene clustering and binary codes computation are performed simultaneously for a query video. Hence, this reduces the retrieval time.

G. Analysis

Comparing the results in terms of mean average precision, we can find that the performances are almost similar for CCV and YLI datasets with the length of bits increasing from 16 to 96. We attribute this to both the dataset and the feature. First, the performance in terms of mean average precision may encounter saturation with the specified length of bits and the number of categories. The numbers of categories in the CCV and YLI datasets are 20 and 10. Note that the length of bits shown in the figures is related to a frame. This means that the length of bits for each video is at least 480 (16*30). Thus, the performance of hashing methods may encounter saturation with the given length of bits and number of categories. This is further validated by the results on the ActivityNet dataset, which contains 200 categories. For such a dataset with several times of categories against the previous two datasets, the performance improvement reduces when increasing the length of bits of 64 to 96 compared against that from 32 to 64. Second, the saturation of performance in terms of mean average precision may be related to the discriminative ability of the original feature. For features with good discriminative ability, it requires longer binary codes to preserve the discrimination. Comparing the results with the CNN features in Figs. 5 and 8 against the results with the handcraft features in Figs. 4 and 7, we can observe performance improvement for some hashing methods with the increase of bit length when provided with more discriminative features.

To provide more discriminative video representations, it is promising to combine the proposed hashing learning framework with the motion extraction. With the motion information captured by the optical flow or RNN feature, we can apply such feature extraction on the clustered scenes of each video. And both the motion information and the frame representation can be utilized to learn hashing codes jointly in a multi-view hashing way, such as [69]. While the proposed hashing leaning framework uses the pre-extracted CNN features as input, the pooling strategies [4], [70] can be combined with it by applying the pooling on the shots of each video scene. Such combination can reduce the computation complexity, eliminate irrelevant frames and enhance the discriminative power. Furthermore, as stated in [71], the semantic representation is comparable to hand-crafted low-level features in performance for event detection. Thus, it is also feasible to apply hashing on the semantic representations to retrieve the semantic neighbor.

V. Conclusions

In this paper, we have proposed a nonlinear structural hashing (NSH) approach to map videos as binary codes for scalable video search. The success of the proposed structure video hashing is attributed to three primary aspects: 1) the exploration of structure information between frames in a video, which is described as the scene in a video and represented by subspace; 2) the preservation of similar relationship between frames of the same scene to achieve the similarity of their binary codes; and 3) the design of multilayer neural network to preserve the nonlinear relationship between videos. Extensive experiments on three large scale datasets fully verify the efficacy of our approach.

There are three interesting directions for future work:

1) How to extend our approach to explore motion information, such as optical flow and RNN, to improve the performance.
2) How to extend our approach to explore more effective video representations, such as semantic representation [71] and video representation with complex pooling strategy [4], [70], to reduce the computation complexity.
3) How to extend our approach to represent videos by more discriminative image features with the integration of extra features, such as audio, motion and text, to further improve the retrieval performance.

References

[1] S. Wei, Y. Zhao, Z. Zhu, and N. Liu, "Multimodal fusion for video search reranking," *IEEE Transactions on Knowledge and Data Engineering*, vol. 22, no. 8, pp. 1191–1199, 2010.

[2] L. Jiang, A. G. Hauptmann, and G. Xiang, "Leveraging high-level and low-level features for multimedia event detection," in *ACM Multimedia Conference*, 2012, pp. 449–458.

[3] A. Habibian, T. Mensink, and C. G. M. Snoek, "Videostory: A new multimedia embedding for few-example recognition and translation of events," in *ACM Multimedia Conference*, 2014, pp. 17–26.

[4] Z. Xu, Y. Yang, and A. G. Hauptmann, "A discriminative CNN video representation for event detection," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 1798–1807.

[5] Y. Yang, Z. Ma, Z. Xu, S. Yan, and A. G. Hauptmann, "How related exemplars help complex event detection in web videos?" in *IEEE International Conference on Computer Vision*, 2013, pp. 2104–2111.

[6] A. Tamrakar, S. Ali, Q. Yu, J. Liu, O. Javed, A. Divakaran, H. Cheng, and H. S. Sawhney, "Evaluation of low-level features and their combinations for complex event detection in open source videos," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2012, pp. 3681–3688.

[7] P. Natarajan, S. Wu, S. N. P. Vitaladevuni, X. Zhuang, S. Tsakalidis, U. Park, R. Prasad, and P. Natarajan, "Multimodal feature fusion for robust event detection in web videos," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2012, pp. 1298–1305.

[8] S. Yu, L. Jiang, Z. Xu, Y. Yang, and A. G. Hauptmann, "Content-based video search over 1 million videos with 1 core in 1 second," in *ACM International Conference in Multimedia Retrieval*, 2015, pp. 419–426.

[9] Y. Li, R. Wang, Z. Huang, S. Shan, and X. Chen, "Face video retrieval with image query via hashing across euclidean space and riemannian manifold," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 4758–4767.

[10] L. Cao, Z. Li, Y. Mu, and S. Chang, "Submodular video hashing: a unified framework towards video pooling and indexing," in *ACM Multimedia Conference*, 2012, pp. 299–308.

[11] G. Ye, D. Liu, J. Wang, and S. Chang, "Large-scale video hashing via structure learning," in *IEEE International Conference on Computer Vision*, 2013, pp. 2272–2279.

[12] J. Wang, J. Sun, J. Liu, X. Nie, and H. Yan, "A visual saliency based video hashing algorithm," in *IEEE International Conference on Image Processing*, 2012, pp. 645–648.

[13] A. Gionis, P. Indyk, and R. Motwani, "Similarity search in high dimensions via hashing," in *International Conference on Very Large Data Bases*, 1999, pp. 518–529.

[14] Y. Weiss, A. Torralba, and R. Fergus, "Spectral hashing," in *Advances in Neural Information Processing Systems*, 2008, pp. 1753–1760.

[15] Y. Gong, S. Lazebnik, A. Gordo, and F. Perronnin, "Iterative quantization: A procrustean approach to learning binary codes for large-scale image retrieval," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 12, pp. 2916–2929, 2013.

[16] R. Salakhutdinov and G. E. Hinton, "Semantic hashing," *International Journal Approximate Reasoning*, vol. 50, no. 7, pp. 969–978, 2009.

[17] W. Liu, J. Wang, S. Kumar, and S. Chang, "Hashing with graphs," in *International Conference on Machine Learning*, 2011, pp. 1–8.

[18] K. He, F. Wen, and J. Sun, "K-means hashing: An affinity-preserving quantization method for learning binary compact codes," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2013, pp. 2938–2945.

[19] Y. Gong, S. Kumar, H. A. Rowley, and S. Lazebnik, "Learning binary codes for high-dimensional data using bilinear projections," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2013, pp. 484–491.

[20] M. Á. Carreira-Perpiñán and R. Raziperchikolaei, "Hashing with binary autoencoders," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 557–566.

[21] M. Rastegari, C. Keskin, P. Kohli, and S. Izadi, "Computationally bounded retrieval," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 1501–1509.

[22] V. E. Liong, J. Lu, G. Wang, P. Moulin, and J. Zhou, "Deep hashing for compact binary codes learning," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 2475–2483.

[23] A. Torralba, R. Fergus, and Y. Weiss, "Small codes and large image databases for recognition," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2008, pp. 1–8.

[24] J. Wang, S. Kumar, and S. Chang, "Semi-supervised hashing for large-scale search," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 12, pp. 2393–2406, 2012.

[25] M. Rastegari, A. Farhadi, and D. A. Forsyth, "Attribute discovery via predictable discriminative binary codes," in *European Conference on Computer Vision*, 2012, pp. 876–889.

[26] M. Norouzi and D. J. Fleet, "Minimal loss hashing for compact binary codes," in *International Conference on Machine Learning*, 2011, pp. 353–360.

[27] M. Norouzi, D. J. Fleet, and R. Salakhutdinov, "Hamming distance metric learning," in *Advances in Neural Information Processing Systems*, 2012, pp. 1070–1078.

[28] C. Strecha, A. M. Bronstein, M. M. Bronstein, and P. Fua, "LDAHash: Improved matching with smaller descriptors," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 1, pp. 66–78, 2012.

[29] J. Wang, W. Liu, A. X. Sun, and Y. Jiang, "Learning hash codes with listwise supervision," in *IEEE International Conference on Computer Vision*, 2013, pp. 3032–3039.

[30] T. Ge, K. He, and J. Sun, "Graph cuts for supervised binary coding," in *European Conference on Computer Vision*, 2014, pp. 250–264.

[31] W. Liu, J. Wang, R. Ji, Y. Jiang, and S. Chang, "Supervised hashing with kernels," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2012, pp. 2074–2081.

[32] G. Lin, C. Shen, Q. Shi, A. van den Hengel, and D. Suter, "Fast supervised hashing with decision trees for high-dimensional data," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 1971–1978.

[33] F. Shen, C. Shen, W. Liu, and H. T. Shen, "Supervised discrete hashing," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 37–45.

[34] M. Kan, D. Xu, S. Shan, and X. Chen, "Semisupervised hashing via kernel hyperplane learning for scalable image search," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 24, no. 4, pp. 704–713, 2014.

[35] Z. Lin, G. Ding, M. Hu, and J. Wang, "Semantics-preserving hashing for cross-view retrieval," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 3864–3872.

[36] F. Zhao, Y. Huang, L. Wang, and T. Tan, "Deep semantic ranking based hashing for multi-label image retrieval," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 1556–1564.

[37] J. Hsieh, S. Yu, and Y. Chen, "Motion-based video retrieval by trajectory matching," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 16, no. 3, pp. 396–409, 2006.

[38] D. Zhong and S. Chang, "An integrated approach for content-based video object segmentation and retrieval," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 9, no. 8, pp. 1259–1268, 1999.

[39] L. Gao, Z. Li, and A. K. Katsaggelos, "An efficient video indexing and retrieval algorithm using the luminance field trajectory modeling," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 19, no. 10, pp. 1566–1570, 2009.

[40] Y. Liu, T. Mei, X. Wu, and X. Hua, "Multigraph-based query-independent learning for video search," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 19, no. 12, pp. 1841–1850, 2009.

[41] Y. Lai and C. Yang, "Video object retrieval by trajectory and appearance," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 25, no. 6, pp. 1026–1037, 2015.

[42] J. L. Bentley, "Multidimensional binary search trees used for associative searching," *Communication of the ACM*, vol. 18, no. 9, pp. 509–517, 1975.

[43] J. K. Uhlmann, "Satisfying general proximity/similarity queries with metric trees," *Information Processing Letters*, vol. 40, no. 4, pp. 175–179, 1991.

[44] P. N. Yianilos, "Data structures and algorithms for nearest neighbor search in general metric spaces," in *ACM/SIGACT-SIAM Symposium on Discrete Algorithms*, 1993, pp. 311–321.

[45] S. Arya, D. M. Mount, N. S. Netanyahu, R. Silverman, and A. Y. Wu, "An optimal algorithm for approximate nearest neighbor searching fixed dimensions," *Journal of the ACM*, vol. 45, no. 6, pp. 891–923, 1998.

[46] H. Jégou, M. Douze, and C. Schmid, "Product quantization for nearest neighbor search," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 1, pp. 117–128, 2011.

[47] Y. Gong, S. Kumar, V. Verma, and S. Lazebnik, "Angular quantization-based binary codes for fast similarity search," in *Advances in Neural Information Processing Systems*, 2012, pp. 1205–1213.

[48] Y. Ma, L. Lu, H. Zhang, and M. Li, "A user attention model for video summarization," in *ACM Multimedia Conference*, 2002, pp. 533–542.

[49] C. Ngo, Y. Ma, and H. Zhang, "Video summarization and scene detection by graph modeling," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 15, no. 2, pp. 296–305, 2005.

[50] K. Sze, K. Lam, and G. Qiu, "A new key frame representation for video segment retrieval," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 15, no. 9, pp. 1148–1155, 2005.

[51] Z. Cernekova, I. Pitas, and C. Nikou, "Information theory-based shot cut/fade detection and video summarization," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 16, no. 1, pp. 82–91, 2006.

[52] Y. Peng and C. Ngo, "Clip-based similarity measure for query-dependent clip retrieval and video summarization," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 16, no. 5, pp. 612–627, 2006.

[53] B. Coskun, B. Sankur, and N. D. Memon, "Spatio-temporal transform based video hashing," *IEEE Transactions on Multimedia*, vol. 8, no. 6, pp. 1190–1208, 2006.

[54] J. Hu, J. Lu, and Y. Tan, "Discriminative deep metric learning for face verification in the wild," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 1875–1882.

[55] B. Kulis and T. Darrell, "Learning to hash with binary reconstructive embeddings," in *Advances in Neural Information Processing Systems*, 2009, pp. 1042–1050.

[56] J. He, W. Liu, and S. Chang, "Scalable similarity search with optimized kernel hashing," in *ACM SIGKDD Conferences on Knowledge Discovery and Data Mining*, 2010, pp. 1129–1138.

[57] R. Wang, S. Shan, X. Chen, Q. Dai, and W. Gao, "Manifold-manifold distance and its application to face recognition with image sets," *IEEE Transactions on Image Processing*, vol. 21, no. 10, pp. 4466–4479, 2012.

[58] Y. Jiang, G. Ye, S. Chang, D. P. W. Ellis, and A. C. Loui, "Consumer video understanding: a benchmark database and an evaluation of human and machine performance," in *ACM International Conference in Multimedia Retrieval*, 2011, p. 29.

[59] J. Bernd, D. Borth, B. Elizalde, G. Friedland, H. Gallagher, L. R. Gottlieb, A. Janin, S. Karabashlieva, J. Takahashi, and J. Won, "The YLI-MED corpus: Characteristics, procedures, and plans," *CoRR*, vol. abs/1503.04250, 2015.

[60] B. Thomee, D. A. Shamma, G. Friedland, B. Elizalde, K. Ni, D. Poland, D. Borth, and L. Li, "YFCC100M: the new data in multimedia research," *Communication of the ACM*, vol. 59, no. 2, pp. 64–73, 2016.

[61] F. C. Heilbron, V. Escorcia, B. Ghanem, and J. C. Niebles, "Activitynet: A large-scale video benchmark for human activity understanding," in *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2015, Boston, MA, USA, June 7-12, 2015*, 2015, pp. 961–970. [Online]. Available: http://dx.doi.org/10.1109/CVPR.2015.7298698

[62] J. Song, Y. Yang, Z. Huang, H. T. Shen, and J. Luo, "Effective multiple feature hashing for large-scale near-duplicate video retrieval," *IEEE Transactions on Multimedia*, vol. 15, no. 8, pp. 1997–2008, 2013.

[63] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, 2004.

[64] K. Mikolajczyk and C. Schmid, "Scale & affine invariant interest point detectors," *International Journal of Computer Vision*, vol. 60, no. 1, pp. 63–86, 2004.

[65] J. Sivic and A. Zisserman, "Video google: A text retrieval approach to object matching in videos," in *IEEE International Conference on Computer Vision*, 2003, pp. 1470–1477.

[66] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *CoRR*, vol. abs/1409.1556, 2014.

[67] J. Deng, W. Dong, R. Socher, L. Li, K. Li, and F. Li, "ImageNet: A large-scale hierarchical image database," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2009, pp. 248–255.

[68] J. Donahue, Y. Jia, O. Vinyals, J. Hoffman, N. Zhang, E. Tzeng, and T. Darrell, "Decaf: A deep convolutional activation feature for generic visual recognition," in *International Conference on Machine Learning*, 2014, pp. 647–655.

[69] D. Zhang, F. Wang, and L. Si, "Composite hashing with multiple information sources," in *Proceeding of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2011, Beijing, China, July 25-29, 2011*, 2011, pp. 225–234.

[70] X. Chang, Y.-L. Yu, Y. Yang, and E. P. Xing, "Semantic pooling for complex event analysis in untrimmed videos," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2016.

[71] X. Chang, Z. Ma, Y. Yang, Z. Zeng, and A. G. Hauptmann, "Bi-level semantic representation analysis for multimedia event detection," *IEEE Transactions on Cybernetics*, 2016.

Zhixiang Chen received the B.S. degree in microelectronics from the Xi'an Jiaotong University, Xi'an, China. He is currently pursuing the Ph.D. degree with the Department of Automation, Tsinghua University, Beijing, China. His current research interests include large scale visual search, binary embedding and hashing learning.

Jiwen Lu received the B.Eng. degree in mechanical engineering and the M.Eng. degree in electrical engineering from the Xian University of Technology, Xian, China, in 2003 and 2006, respectively, and the Ph.D. degree in electrical engineering from Nanyang Technological University, Singapore, in 2012. From 2011 to 2015, he was a Research Scientist with the Advanced Digital Sciences Center, Singapore. He is currently an Associate Professor with the Department of Automation, Tsinghua University, Beijing, China. His current research interests include computer vision, pattern recognition, and machine learning. He has authored or co-authored over 150 scientific papers in these areas, where 40 were the IEEE Transactions papers. He is the Workshop Chair/Special Session Chair/Area Chair for over ten international conferences. He was a recipient of the National 1000 Young Talents Plan Program in 2015. He serves as an Associate Editor of the Pattern Recognition Letters, the Neurocomputing, and the IEEE ACCESS, a Guest Editor for Special Issue of 5 journals including Pattern Recognition, Computer Vision and Image Understanding, and Image and Vision Computing, and an Elected Member of the Information Forensics and Security Technical Committee of the IEEE Signal Processing Society.

Jianjiang Feng is an associate professor in the Department of Automation at Tsinghua University, Beijing. He received the B.S. and Ph.D. degrees from the School of Telecommunication Engineering, Beijing University of Posts and Telecommunications, China, in 2000 and 2007. From 2008 to 2009, he was a Post Doctoral researcher in the PRIP lab at Michigan State University. He is an Associate Editor of Image and Vision Computing. His research interests include fingerprint recognition and computer vision.

Jie Zhou received the B.S. and M.S. degrees from the Department of Mathematics, Nankai University, Tianjin, China, in 1990 and 1992, respectively, and the Ph.D. degree from the Institute of Pattern Recognition and Artificial Intelligence, Huazhong University of Science and Technology, Wuhan, China, in 1995. From 1995 to 1997, he served as a Post-Doctoral Fellow with the Department of Automation, Tsinghua University, Beijing, China. Since 2003, he has been a Full Professor with the Department of Automation, Tsinghua University. In recent years, he has authored over 100 papers in peer-reviewed journals and conferences. Among them, over 40 papers have been published in top journals and conferences, such as the IEEE PAMI, TIP, and CVPR. His current research interests include computer vision, pattern recognition, and image processing. He received the National Outstanding Youth Foundation of China Award. He is an Associate Editor of the International Journal of Robotics and Automation and two other journals.